1. What are Accessors, mutators, and @property?

Ans:

Accessor Method:

This method is used to access the state of the object i.e., the data hidden in the object can be accessed from this method. However, this method cannot change the state of the object, it can only access the data hidden. We can name these methods with the word get. Mutator Method:

This method is used to mutate/modify the state of an object i.e., it alters the hidden value of the data variable. It can set the value of a variable instantly to a new value. This method is also called the update method. Moreover, we can name these methods with the word set.

2. Differentiate between append () and extend () methods.?

Append ()	Extend ()
To add a single entry to the end of a list, use the append () function.	To add additional elements or an iterable to the end of a list, use the extend () function.
Accepts only one input element.	Accepts as input an iterable (such as a list or tuple).
The append () function adds the full input to the list as a single item.	Extend () adds each item to the list independently after iterating through each one in the input.
Append has consistent time complexity.	Extend has a time complexity of O(k). Where k is the length of the list which needs to be added.
Since append () only executes one operation, it is typically quicker and more effective than extend().	When adding elements from numerous iterables or with huge inputs, extend () could take longer.

3. Name a few methods that are used to implement Functionally Oriented Programming in Python?

Ans: Functional programming languages focus on declarations and expressions rather than the execution of statements. Python offers some built-in functions

1) map ()

2) filter ()

3) reduce ()

4. What is the output of the following?

x = [‘ab’, ‘cd’]

print (len(map(list, x)))

Ans: This will show an error

The error is an “object of type 'map' has no len()”

5. What is the output of the following?

x = [‘ab’, ‘cd’]

print(len(list(map(list, x)))

Ans: The output is 2

6. Explain a few methods to implement Functionally Oriented Programming in Python

Ans: Python offers two built-in functions, map() and filter(), that fit the functional programming paradigm. A third, reduce(), is no longer part of the core language but is still available from a module called functools. Each of these three functions takes another function as one of its arguments.

1) map ():

The first function on the docket is map(), which is a Python built-in function. With map(), you can apply a function to each element in an iterable in turn, and map() will return an iterator that yields the results. This can allow for some very concise code because a map() statement can often take the place of an explicit loop.

2) filter ():

filter() allows you to select or filter items from an iterable based on evaluation of the given function.

3) reduce ():

reduce() applies a function to the items in an iterable two at a time, progressively combining them to produce a single result.

7. Explain database connection in Python Flask?

Flask is a micro web framework written in python. Micro-framework is normally a framework with little to no dependencies on external libraries. Though being a micro framework almost everything can be implemented using python libraries and other dependencies when and as required.

Installing Flask

In any directory where you feel comfortable create a folder and open the command line in the directory. Create a python virtual environment using the command below.

```
python -m venv <name>
```

Once the command is done running activate the virtual environment using the command below.

```
<name>\scripts\activate
```

Now, install Flask using pip (package installer for python). Simply run the command below.

```
pip install Flask
```

Creating app.py

Once the installation is done create a file name app.py and open it in your favourite editor. To check whether Flask has been properly installed you can run the following code.

```
from flask import Flask

app = Flask(__name__)

@app.route('/')

def check():

   return 'Flask is working'

if __name__ == '__main__':

   app.run()
```

Setting Up SQLAlchemy

Now, let's move on to creating a database for our application. For the purpose of this article, we will be using SQLAlchemy a database toolkit, and an ORM(Object Relational Mapper). We will be using pip again to install SQLAlchemy. The command is as follows,

pip install flask-sqlalchemy

In your app.py file import SQLAlchemy as shown in the below code. We also need to add a configuration setting to our application so that we can use SQLite database in our application. We also need to create an SQLAlchemy database instance which is as simple as creating an object.

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy


app = Flask(__name__)
app.debug = True


# adding configuration for using a sqlite database
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///site.db'


# Creating an SQLAlchemy instance
db = SQLAlchemy(app)


if __name__ == '__main__':
   app.run()
```

8. Write a Python function that checks whether a passed string is palindrome or not?

Note: A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

```
def isPalindrome(s):
        return s == s[::-1]


s = "malayalam"
ans = isPalindrome(s)
if ans:
   print("Yes")
else:
   print("No")
```

9. Write a Python program to calculate the sum of a list of numbers.

```
total = 0
# creating a list
list1 = [11, 5, 17, 18, 23]
for i in range(0, len(list1)):
   total = total + list1[i]
# printing total value
print("Sum of all elements in given list: ", total)
```

10. How to retrieve data from a table in MySQL database through Python code? Explain.

READ Operation on any database means to fetch some useful information from the database. You can fetch data from MYSQL using the fetch() method provided by the mysql-connector-python.

The cursor.MySQLCursor class provides three methods namely fetchall(), fetchmany() and, fetchone() where,

The fetchall() method retrieves all the rows in the result set of a query and returns them as list of tuples. (If we execute this after retrieving few rows it returns the remaining ones).

The fetchone() method fetches the next row in the result of a query and returns it as a tuple.

The fetchmany() method is similar to the fetchone() but, it retrieves the next set of rows in the result set of a query, instead of a single row.

Note − A result set is an object that is returned when a cursor object is used to query a table.

rowcount − This is a read-only attribute and returns the number of rows that were affected by an execute() method.

Example

Following example fetches all the rows of the EMPLOYEE table using the SELECT query and from the obtained result set initially, we are retrieving the first row using the fetchone() method and then fetching the remaining rows using the fetchall() method.

```
import mysql.connector

#establishing the connection
conn = mysql.connector.connect(
  user='root', password='password', host='127.0.0.1', database='mydb')

#Creating a cursor object using the cursor() method
cursor = conn.cursor()

#Retrieving single row
sql = '''SELECT * from EMPLOYEE'''

#Executing the query
cursor.execute(sql)

#Fetching 1st row from the table
result = cursor.fetchone();
print(result)

#Fetching 1st row from the table
result = cursor.fetchall();
print(result)

#Closing the connection
conn.close()
```

11. Write a Python program to read a random line from a file.

```
import random
def random_line(fname):
   lines = open(fname).read().splitlines()
   return random.choice(lines)
print(random_line('test.txt'))
```

12. Write a Python program to count the number of lines in a text file.

ANS: Python counts the number of lines in a string using read lines ()

The read lines () are used to read all the lines in a single go and then return them as each line is a string element in a list. This function can be used for small files.

```
with open (r"myfile.txt", 'r') as fp:
   lines = len (fp. read lines ())
   print ('Total Number of lines:', lines)
```

13. Name few Python modules/libraries for Statistical, Numerical, and scientific computations?

ANS: Python is widely used in scientific and numeric computing:

1. NumPy: The fundamental package for numerical computations in Python. It provides support for large, multi-dimensional arrays and matrices, along with a variety of mathematical functions to operate on these arrays.

2. SciPy: Built on top of NumPy, SciPy is a library for scientific and technical computing. It includes modules for optimization, linear algebra, integration, interpolation, signal and image processing, and more.

3. pandas: A versatile data manipulation and analysis library. It provides data structures like Data Frames for handling tabular data and Series for one-dimensional data, making data analysis and manipulation easier.

4. matplotlib: A 2D plotting library that produces publication-quality figures and charts. It provides a variety of plot types and customization options to visualize data effectively.

5. Seaborn: Built on top of matplotlib, Seaborn is a higher-level data visualization library that offers an easy-to-use interface for creating attractive and informative statistical graphics.

6. Stats models: A library for estimating and interpreting statistical models. It includes various classes and functions for performing statistical analyses, including linear regression, ANOVA, time series analysis, and more.

7. scikit-learn: A machine learning library that provides tools for classification, regression, clustering, dimensionality reduction, and more. It's built on NumPy, SciPy, and matplotlib and is widely used for machine learning tasks.

8. SymPy: A symbolic mathematics library that allows you to perform symbolic mathematics computations. It can manipulate algebraic expressions, solve equations, perform calculus operations, and more.

9. Astropy: A library for astronomy-related computations. It provides tools for handling celestial coordinates, time and date calculations, units and quantities, and more.

10. NetworkX: A library for the creation, manipulation, and study of complex networks and graphs. It's useful for analyzing relationships and structures in various fields like social networks, biology, and transportation systems.

11. BioPython: Specifically for computational biology and bioinformatics tasks. It provides tools for parsing biological data formats, performing sequence analysis, protein structure analysis, and more.

12. OpenCV: A computer vision library that offers a wide range of functions for image and video analysis, including object detection, image processing, feature extraction, and more.

These are just a few of the many Python libraries available for statistical, numerical, and scientific computations.

14. What does this mean: *args, **kwargs? And why would we use it?

ANS: *args means if give * Infront of parameters than if we give any arguments in function call it will take like tuple or list or set. In function if we give ** before args than all values what all are in function arguments it will convert into dictionary. When we are writing Python code, functions are something that we cannot overlook. Functions are beneficial in removing the repetition of code and making it modular . Functions generally need some data to work on, and it can be in the form of strings, numbers, or even other functions. In the programming world, this data is referred to as arguments.

15. What are negative indexes and why are they used?

ANS: As we know, indexes are used in arrays in all the programming languages. We can access the elements of an array by going through their indexes. But no programming language allows us to use a negative index value such as -4. Python programming language supports negative indexing of arrays, something which is not available in arrays in most other programming languages. This means that the index value of -1 gives the last element, and -2 gives the second last element of an array. The negative indexing starts from where the array ends. This means that the last element of the array is the first element in the negative indexing which is -1.

16. How can you randomize the items of a list in place in Python?

ANS: The method shuffle () can be used to randomize the items of a list in place. It should be noted that this function is not accessible directly and therefore we need to import or call this function using random static object.

Syntax: shuffle (lst)

Here, 'lst' is passed as a parameter which could be a list or tuple. The shuffle() returns a reshuffled list of items.

Example: import random

list = [20, 16, 10, 5];

random. Shuffle(list)

print "Reshuffled list: ", list

random. Shuffle(list)

print "Reshuffled list: ", list

Output:

Reshuffled list: [16, 10, 5, 20]

Reshuffled list: [5, 20, 10, 16]

17. Write a sorting algorithm for a numerical dataset in Python.

Ans :

```
def bubblesort(list):

# Swap the elements to arrange in order

    for iter_num in range(len(list)-1,0,-1):

    for idx in range(iter_num):

    if list[idx]>list[idx+1]:

        temp = list[idx]

        list[idx] = list[idx+1]

        list[idx+1] = temp
```

```
list = [19,2,31,45,6,11,121,27]

bubblesort(list)

print(list)
```

Output :

[2, 6, 11, 19, 27, 31, 45, 121]

18. Looking at the below code, write down the final values of A0, A1, …An.

```
A0 = dict(zip(('a','b','c','d','e'),(1,2,3,4,5)))
A1 = range(10)
A2 = sorted([i for i in A1 if i in A0])
A3 = sorted([A0[s] for s in A0])
A4 = [i for i in A1 if i in A3]
A5 = {i:i*i for i in A1}
A6 = [[i,i*i] for i in A1]
print(A0,A1,A2,A3,A4,A5,A6)
```

Ans :

```
A0 = {'a': 1, 'c': 3, 'b': 2, 'e': 5, 'd': 4}  # the order may vary

A1 = range(0, 10) # or [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] in python 2
```

A2 = []

A3 = [1, 2, 3, 4, 5]

A4 = [1, 2, 3, 4, 5]

A5 = {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}

A6 = [[0, 0], [1, 1], [2, 4], [3, 9], [4, 16], [5, 25], [6, 36], [7, 49], [8, 64], [9, 81]]

19. What is map function in Python?

Ans : Map in Python is a function that works as an iterator to return a result after applying a function to every item of an iterable (tuple, lists, etc.). It is used when you want to apply a single transformation function to all the iterable elements. The iterable and function are passed as arguments to the map in Python.

The syntax of the Python map() function is:

map(function, iterables)

In the above syntax:

- function: It is the transformation function through which all the items of the iterable will be passed.
- iterables: It is the iterable (sequence, collection like list or tuple) that you want to map.

Eg.

```
# Defining a function
def mul(i):
    return i * i
# Using the map function
```

```
x = map(mul, (3, 5, 7, 11, 13))
print (x)
print(list(x))
```

Output :

```
[9, 25, 49, 121, 169]
```

20. How to get indices of N maximum values in a NumPy array?

Ans : We can get the indices of the N maximum values in a NumPy array using the numpy.argsort() function.This function returns the indices that would sort the array in ascending order. By using this function along with slicing, you can get the indices of the N maximum values.

Example :

```
import numpy as np

def get_n_max_indices(arr, n):
    sorted_indices = np.argsort(arr)[::-1]
    return sorted_indices[:n]
array = np.array([10, 5, 8, 15, 3, 20, 12])
n = 3
max_indices = get_n_max_indices(array, n)
print("Indices of", n, "maximum values:", max_indices)
```

Output :

```
Indices of 3 maximum values: [5 3 6]
```

21. How do you calculate percentiles with Python/ NumPy?

Ans :

Input :

```
import numpy as np
data = np.array([12, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60])
percentile_25 = np.percentile(data, 25)
percentile_75 = np.percentile(data, 75)
print("25th percentile:", percentile_25)
print("75th percentile:", percentile_75)
```

Output :

```
25th percentile: 25.0
75th percentile: 45.0
```